Programação Visual
Trabalho de Laboratório nº 3

Objetivo	ASP.NET Core MVC – Introdução. Aplicações básicas
Programa	Pretende-se criar um *Site* para uma empresa de alojamento local para o IPS. O *site* deverá permitir a visualização dos quartos disponíveis e os seus detalhes.
Regras	Implementar o código necessário e testar no fim de cada nível. Use as convenções de codificação adotadas para a linguagem C# e para o modelo MVC.

Nível 1

- Crie uma aplicação com o *template* **ASP.NET Core Web App (Model-View-Controller)** com o nome **SleepSetubal**.
- Comece por editar a página de Layout que serve de base ao *site* e efetue as seguintes alterações:
  - Nas opções de menu do *site* substitua a palavra "**Privacy"** por **"Privacidade".**
  - Elimine a linha que tem a opção de menu **"Home".**
- Dentro da pasta **wwwroot** crie uma pasta **images** e adicione a imagem **under_construction.png** fornecida nos materiais do laboratório. Na **View** *index* (**index.cshtml**) do controlador **Home** substitua todo o conteúdo *html* por um parágrafo com o texto "**A SleepSetubal está em construção**". Inclua, também, a imagem que acrescentou no início.
- Na vista "**Privacy"** substituía o texto em inglês por um texto equivalente em português.
- No controlador **`HomeController`** crie a ação **`Contact`** e a vista associada. Na criação da vista use a **Razor View – Empty**. Na **View contact.cshtml** criada adicione o texto que está no ficheiro html no ficheiro "**Contact**" dos materiais fornecidos.
  Para aceder à página criada, inclua mais uma opção de menu na página de *layout*, neste caso a opção "**Contactos**". Coloque-a antes da opção "**Privacidade"**.
- À semelhança do que fez anteriormente, crie a ação **`About`** e a vista associada dentro do mesmo controlador e coloque na vista um texto com o nome(s) do(s) elemento(s) do grupo que está(ão) a desenvolver a página. A ação criada deve ter uma entrada "**Sobre**" no menu principal antes da opção "**Contactos**" criada anteriormente.

Programação Visual
Trabalho de Laboratório nº 3

Nível 2

- Crie uma classe **`Room`** com propriedades para o **Name**, **Type**, **Beds** (Lista de nomes), **Price** (inteiro) e **Available** (booleano).
- Adicione à classe do controlador **Home** um atributo **`rooms`** do tipo **`List<Room>`**. No construtor inicialize a lista de quartos com o código que está no ficheiro **quartos.txt.**
- Crie uma ação **`RecommendedRoom`** e no corpo do método dessa ação, obtenha o primeiro quarto da lista de quartos e passe-o depois como argumento do método **View**. Crie a chave **"`IndexRoom`"** do **`ViewData`** colocando como valor um texto com o índice do quarto e o total de quartos na lista (por exemplo "**Quarto 1 de 4**").
- Para visualizar o quarto recomendado será necessária uma nova **Razor View.** Sendo assim adicione uma nova **View** para a ação **`RecommendedRoom`**. Na vista criada deve utilize o *template* **details.** como modelo use a classe **`Room`**. Altere a vista para que apareça apenas o nome, o tipo e o preço do quarto por noite. No fim adicione o texto recebido no **`ViewData`**.
- Acrescente agora na página de *layout* mais uma opção ao menu do site criado com o texto **"Quarto recomendado"** que deve ir para a vista que acabou de criar.

Nível 3

- Crie um controlador **Room** (Selecione **MVC Controller-Empty** na janela de criação do controlador). Para efeitos de teste, crie um atributo **rooms** do tipo **`List<Room>`** tal como fez para o controlador **Home** e inicie a lista no construtor da classe com o mesmo código utilizado anteriormente.
- Defina agora uma ação **`ListRooms`** que passa para a vista associada a lista de quartos guardada no atributo. Crie a vista associada usando o *template* "**Empty – without model"**. Use depois, na vista, como modelo, o tipo **`IEnumerable<Room>`** de acordo com o objeto passado. Na vista, liste os quartos da lista.
- Inclua no menu principal uma opção "**Quartos"** que vai chamar a ação criada neste nível. Teste e verifique se está a funcionar corretamente.

Nível 4

- Crie uma ação **AvailableRooms** no controlador "**Room"**. Esta ação deve filtrar os quartos (usando LinQ) selecionando apenas os que estão disponíveis e depois passar o resultado para a vista. Na criação da vista irá usar *Scaffolding.*
  Sendo assim, adicione uma vista e na janela de criação da vista escolha o *template* **List** e a classe de modelo **`Room`**. Crie uma opção de menu "Quartos disponíveis**"** para esta ação e verifique o resultado.
- Como deve verificar aparecem na vista uma listagem dos quartos disponíveis. Analise o ficheiro da vista, adicione as camas disponíveis por quarto e remova o que for necessário para que não apareça a coluna **Available** e os links que estão depois. Remova também o link "**Create New"**. Verifique que a vista está agora de acordo com o pretendido.
- Altere a vista para que apareça uma mensagem "**Não existem quartos disponíveis"** quando a lista de quartos estiver vazia.

Programação Visual
Trabalho de Laboratório nº 3

Nível 5

- Crie uma pasta **Documents** dentro da pasta **wwwroot** e adicione os ficheiros "pdf" que estão na pasta dos materiais.
- Adicione agora ao controlador **Home** uma ação com o nome **`MakeReservation`**. Esta ação que retorna um **`VirtualFileResult`** deve retornar em vez da **`View`**, um **`File`**, fornecendo-se o nome do ficheiro criado da seguinte forma:

```
return File("~/documents/nomedoficheiro.pdf", "application/pdf",
"nomedoficheiro.pdf");
```

- Para usar esta ação crie o seguinte *link* na *view* do quarto recomendado:

```
<h4><a asp-controller="Home" asp-action="MakeReservation" class="label
label-success">Reservar</a></h4>
```

- Teste a última funcionalidade para verificar se está a funcionar corretamente no quarto recomendado.

Desafio

- Generalize o método **MakeReservation** para que seja possível reutilizar para todos os quartos e adicione essa função nas vistas dos quartos disponíveis e na vista dos quartos.
- Remover link de reserva nos quartos não disponiveis.

Notas

Para os identificadores siga as convenções adotadas pelo C#, nomeadamente:

- A notação camelCase para o nome das variáveis locais e identificadores privados.
- A notação PascalCase para os nomes públicos dos métodos, classes e interfaces.
- Não utilize o símbolo '_' nos identificadores nem abreviaturas